



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/874,170	06/04/2001	Vasanth Bala	10003355-1	7644

7590 03/02/2006

HEWLETT-PACKARD COMPANY  
Intellectual Property Administration  
P.O. Box 272400  
Fort Collins, CO 80527-2400

EXAMINER

PROCTOR, JASON SCOTT

ART UNIT	PAPER NUMBER
----------	--------------

2123

DATE MAILED: 03/02/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

<b>Office Action Summary</b>	<b>Application No.</b> 09/874,170	<b>Applicant(s)</b> BALA ET AL.	
	<b>Examiner</b> Jason Proctor	<b>Art Unit</b> 2123	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

### Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

### Status

- 1) ☒ Responsive to communication(s) filed on 22 August 2005.
- 2a) ☐ This action is FINAL.                      2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

### Disposition of Claims

- 4) ☒ Claim(s) 1-24 is/are pending in the application.
- 4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.
- 5) ☐ Claim(s) \_\_\_\_\_ is/are allowed.
- 6) ☒ Claim(s) 1-24 is/are rejected.
- 7) ☐ Claim(s) \_\_\_\_\_ is/are objected to.
- 8) ☐ Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

### Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on \_\_\_\_\_ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.  
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).  
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

### Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All    b) ☐ Some \* c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
  2. ☐ Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.
  3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

\* See the attached detailed Office action for a list of the certified copies not received.

### Attachment(s)

- |  |   |
|--|---|
| 1) <input type="checkbox"/> Notice of References Cited (PTO-892)   | 4) <input type="checkbox"/> Interview Summary (PTO-413)<br>Paper No(s)/Mail Date. _____ |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948)                                   | 5) <input type="checkbox"/> Notice of Informal Patent Application (PTO-152)             |
| 3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)<br>Paper No(s)/Mail Date _____ | 6) <input type="checkbox"/> Other: _____  |

### **DETAILED ACTION**

A request for continued examination under 37 CFR 1.114, including the fee set forth in 37 CFR 1.17(e), was filed in this application after final rejection. Since this application is eligible for continued examination under 37 CFR 1.114, and the fee set forth in 37 CFR 1.17(e) has been timely paid, the finality of the previous Office action has been withdrawn pursuant to 37 CFR 1.114. Applicant's submission filed on 22 August 2005 has been entered.

Claims 1-24 were rejected in office action dated 16 May 2005. Claims 1, 8-9, 14, 17-18, and 23 have been amended by Applicants' response dated 22 August 2005. Claims 1-24 have been submitted for reconsideration.

Claims 1-24 have been rejected.

#### ***Claim Objections***

The previous objection to claim 21 has been withdrawn in light of the corrected claim listing. However, Applicants' are encouraged to fully comply with 37 CFR 1.121 when amending the claims, even when correcting minor typographical errors.

#### ***Claim Interpretation***

1. Claims 1, 9, 14, 18, and 23 are objected to because of the following informalities: the limitation "wherein a granularity of the code segment is dynamically tailored to the client to balance server-side and client-side execution and client-side storage requirements, and is

Art Unit: 2123

configured based on predicted code segment usage or prior code segment usage history” cannot be interpreted with confidence.

It is unclear how the functional language “is dynamically tailored to the client to balance server-side and client-side execution and client-side storage requirements” limits the claimed inventions in compliance with 35 U.S.C. § 112, second paragraph, except as meaning “wherein a granularity of the code segment is configured based on predicted code segment usage or prior code segment usage history.” The meaning of “dynamically tailored ... to balance server-side and client-side execution” etc. is indefinite when considered alone and impossible to compare against the prior art. Does the term “balance” in this limitation mean exactly half of the execution is client-side and exactly half is server-side? What are the definite boundaries of a “dynamically tailored” granularity of a code segment?

Therefore, the amended limitations in independent claims 1, 9, 14, 18, and 23, reciting “wherein a granularity of the code segment is dynamically tailored to the client to balance server-side and client-side execution and client-side storage requirements, and is configured based on predicted code segment usage or prior code segment usage history” is interpreted as “wherein a size of the code segment is configured based on predicted code segment usage or prior code segment usage history.” This interpretation is supported by the specification, in particular the disclosure regarding a “granularity” of a code segment (specification, page 13, line 20 – page 15, line 4).

Appropriate correction or clarification is required.

***Claim Rejections - 35 USC § 112***

The following is a quotation of the first paragraph of 35 U.S.C. § 112:

The specification shall contain a written description of the invention, and of the manner and process of making and using it, in such full, clear, concise, and exact terms as to enable any person skilled in the art to which it pertains, or with which it is most nearly connected, to make and use the same and shall set forth the best mode contemplated by the inventor of carrying out his invention.

2. Claims 1-24 are rejected under 35 U.S.C. 112, first paragraph, as failing to comply with the written description requirement. The claim(s) contains subject matter which was not described in the specification in such a way as to reasonably convey to one skilled in the relevant art that the inventor(s), at the time the application was filed, had possession of the claimed invention.

Claims 1, 9, 14, 18, and 23 have been amended to recite “wherein a granularity of the code segments is dynamically tailored to the client to balance server-side and client-side execution and client-side storage requirements, and is configured based on predicted code segment usage or prior code segment usage history” which was not described in the specification as originally filed. The specification attempts to incorporate by reference three non-patent publications to describe and provide enabling support for this limitation (specification, page 13, line 20 – page 14, line 3).

Accordingly, the subject matter that provides adequate written description and enabling support for this claimed feature must be inserted by amendment into the body of the specification in order to provide a complete disclosure of the invention. Please see MPEP 608.01(p).

***Claim Rejections - 35 USC § 103***

The following is a quotation of 35 U.S.C. § 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

The factual inquiries set forth in *Graham v. John Deere Co.*, 383 U.S. 1, 148 USPQ 459 (1966), that are applied for establishing a background for determining obviousness under 35 U.S.C. § 103(a) are summarized as follows:

1. Determining the scope and contents of the prior art.
2. Ascertaining the differences between the prior art and the claims at issue.
3. Resolving the level of ordinary skill in the pertinent art.
4. Considering objective evidence present in the application indicating obviousness or nonobviousness.

This application currently names joint inventors. In considering patentability of the claims under 35 U.S.C. § 103(a), the examiner presumes that the subject matter of the various claims was commonly owned at the time any inventions covered therein were made absent any evidence to the contrary. Applicant is advised of the obligation under 37 CFR 1.56 to point out the inventor and invention dates of each claim that was not commonly owned at the time a later invention was made in order for the examiner to consider the applicability of 35 U.S.C. § 103(c) and potential 35 U.S.C. § 102(e), (f) or (g) prior art under 35 U.S.C. § 103(a).

1. Claims 1-20 are rejected under 35 U.S.C. § 103(a) as being unpatentable over US Patent No. 6,370,687 to Shimura (supplied by Applicant).

Art Unit: 2123

Regarding claim 1, Shimura teaches a network computer system (column 2, lines 16-30) that comprises:

a server (Fig. 1, reference 10) coupled to the network (Fig. 1, reference 16),

an application code source (Web server from which programs are retrieved, column 4, lines 27-35; Fig. 1, reference 20),

a server code manager coupled to the application code source (substitute compile server, column 4, lines 35-42; Fig. 1, reference 10) and

a client which does not have application code and is coupled to the network (column 4, lines 27-35).

It would be obvious to a person of ordinary skill in the art at the time of Applicants' invention, in combination with his own knowledge of the particular art, that these clients should have a CPU for executing native code. Official notice is taken that it is well known in the art to provide a CPU with a code cache. Applicants have not traversed this use of Official Notice and as such, it is regarded as admitted prior art. See MPEP 2144.03 (C).

Shimura teaches that the method used by this system comprises:

a client requesting the client application from the substitute compile server (column 4, lines 35-42),

the server code manager (substitute compile server) requests the program from the application code source (Web server) (column 4, lines 35-42),

the server code manager compiles the program into a form usable by the client (column 4, 42-48),

the size of the code segments is configured based on predicted code segment usage or prior code segment usage history [*“By virtue of compilation and caching based on the prediction of the next Java program in the substitute compile server...”* (column 8, lines 6-13); *“a Java program to be requested next to the requested Java program is predicted and is previously compiled.”* (column 7, lines 37-41); etc.]);

transmitting the program to the client for native execution (column 4, lines 42-48), and requesting further code from the server code manager as necessary (column 8, lines 6-18).

Shimura’s teaching regarding class files (column 7, line 42 – column 8, line 18) would be obvious to a person of ordinary skill in the art at the time of Applicants’ invention as functionally equivalent to “code segments”. Additionally, Shimura explicitly teaches the suitability of this method to compiling on a class-by-class basis, or performing translation and optimization on a method-by-method basis (column 9, line 63 – column 10, line 9).

The claimed feature of storing the application code on the server code manager, contrasted with Shimura’s teaching of the application code source (Web server) as external to the server code manager (substitute compile server) is regarded as an obvious rearrangement of parts and therefore not a patentable distinction over the prior art. Additionally, Shimura teaches an alternative scenario wherein the code is stored on the server (column 6, lines 2-7).

The system and method taught by Shimura therefore renders obvious the system of claim 1.

In response, Applicants argue primarily that:



Art Unit: 2123

Applicant respectfully states the Claims 1, 9, 14, 18 and 23 include the feature “storing code segments on the client device.” Support for the Claimed feature can be found throughout the Specification including Figure 2B and the associated discussion.

[...] Shimura does not teach storing the class files on the client CPU. Instead, Applicant understands Shimura to teach storing the class files on the substitute compiler server thereby being available for a plurality of clients.

[...] That is, Applicant understands Shimura to teach a server execution model in combination with a virtual machine model as described in the background section of the present application.

The Examiner respectfully traverses this argument as follows.

The Examiner observes that none of claims 1, 9, 14, 18, and 23 recite “storing code segments on the client device,” however such a feature may be inferred from a combination of the limitations. Claim 9, however, presents no limitations or capability for storing code segments on the client device. Claims 14, 18, and 23 recite limitations such as “storing the code segment in the code cache [coupled to the CPU]” but this is the inherent function of a code cache and does not distinguish the claimed code cache from a prior art code cache. The specific limitations to which Applicants’ arguments refer are unclear. Additionally, a statement that “[s]upport for the Claimed feature can be found throughout the Specification including Figure 2B and the associated discussion” is so broad that the Examiner cannot reasonably ascertain to what specific support Applicants are referring.

As to the second point, the Examiner cannot refute that Shimura does not teach storing the class files on the client CPU. A CPU is not generally referred to or employed as a storage device. However, Shimura clearly teaches that code segments are transferred to the client for execution [“*At the same time, the compiled Java program is returned via the LAN interface 30 to the requesting client 14-1 in which the Java program comprised of the compiled native code is run.*” (column 5, lines 58-61); etc.]. The execution of code segments inherently requires storing

Art Unit: 2123

the code segments on the client in a code cache or some other storage. The client cannot execute code segments that it has not, for at least an instant, stored.

The Examiner cannot refute that Shimura does not teach storing the class files on the client, but does not understand the relevance of that observation. The class files of Shimura are compiled to produce code segments [*“[I]n response to a request ... from a client ... the substitute compile server 10 returns to the client the Java program which has been compiled and optimized into a native code conforming to the execute form of the requester client.”* (column 6, lines 25-30)]. As shown above, those code segments are transferred to the client for execution. None of the claims appear to require “storing class files on the client.”

Additionally, the similarities between Shimura and the disclosed invention of the present application are noted, especially regarding the Java virtual machine features:

As will be discussed in greater detail below, application code source 18 can be ASCII source code, a native code format (either in the format required by client 14 or another format), or a **virtual machine format**. (Specification, page 16, line 26 – page 17, line 1)

Finally, the Examiner cannot properly respond to Applicants’ argument that Shimura discloses what is described in the background section of the application except to state that, from all appearances, Shimura renders the claimed invention obvious.

Applicants’ arguments have been fully considered but have been found unpersuasive.

Applicants further argue that:

Applicant does not understand the teachings of Shimura to address or make obvious the dynamic tailoring of the code segment per client to balance server-side and client-side execution and client side storage requirements. That is, Applicant understands Shimura to utilize the substitute compiler server as a proxy server having a function of compiling JAVA program in the form of a virtual machine computer program provided by the web server in response to a request from the client.

The Examiner respectfully traverses this argument as follows.

The Examiner has given this claim limitation a broad, reasonable interpretation within the confines of 35 U.S.C. § 112, second paragraph, as set forth above. The specific meaning of “balanc[ing] server-side and client-side execution and client side storage requirements” is vague and indefinite when considered in isolation. However, Shimura meets the aforementioned interpretation of this language by teaching both the compilation of a Java program which has been “compiled and optimized into a native code conforming to the execute form of the requester client” (column 6, lines 25-30) and by teaching that the next Java program is “predicted and is previously compiled” (column 7, lines 37-41); etc.). Therefore, inasmuch as the claim limitations comply with 35 U.S.C. § 112, second paragraph, Shimura renders the claimed invention obvious.

Applicants’ arguments have been fully considered but have been found unpersuasive.

Regarding claim 2, Shimura teaches an application code transformation manager for transforming the client application from a first format to a native binary format compatible with a native instruction set of the CPU of the client (substitute compile server and compile controller, column 5, lines 45-61). Shimura teaches a code segment manager for partitioning an application program into segments for transmitting to the client via the network (column 7, line 57 – column 8, line 18).

Regarding claim 3, Shimura teaches that the first format is other than the native execution format of the CPU of the client (column 5, lines 15-26). A compiler is functionally equivalent to

Art Unit: 2123

a “transformation engine to transform the client application from the first format to the native binary format of the CPU of the client”.

Regarding claim 4, Shimura does not explicitly teach that the first format is a source code text format of a programming language and the transformation manager comprises a compiler that compiles and links the client application into a native binary format of the CPU of the client. However, Shimura does explicitly teach that the first format is a “Java program in the form of the virtual machine computer program prepared as an applet on the web page” (column 4, lines 28-30) which is compiled using a Java™ compiler (column 4, lines 42-51 and throughout). It would have been obvious to a person of ordinary skill in the art that the term “Java applet” commonly refers to source code in a text format intended for use in a web page and that source code in a text format is well known input to a typical compiler. It therefore would have been obvious to a person of ordinary skill in the art to implement Shimura’s system where the first format is a source code text format of a programming language and compiling that source code into a native binary format of the CPU of the client.

Regarding claim 5, Shimura teaches that the transformation manager comprises a just-in-time compiler (column 5, lines 8-15).

Regarding claim 6, Shimura’s teaching regarding class files (column 7, line 42 – column 8, line 18) would be obvious to a person of ordinary skill in the art at the time of Applicants’ invention as functionally equivalent to “code segments”. It would be obvious to a person of

ordinary skill in the art at the time of Applicants' invention to implement this functionality with a client code manager that requests needed segments from the server and to branch into the received code segment. Indeed, this is the functionality implied by Shimura (column 7, line 57 – column 8, line 13) although the obvious details of implementation are omitted.

Regarding claim 7, Shimura does not explicitly recite the steps of adjusting branch instructions to link into and out of received code segments as recited. Shimura implies this functionality (column 7, line 42 – column 8, line 18; column 9, line 63 – column 10, line 9). Official notice is taken that the need to link code that is compiled in sections is well known. It would have been obvious to a person of ordinary skill in the art at the time of Applicants' invention, in combination with his own knowledge of the particular art, to adequately support linking sections of compiled code by adjusting the branch instructions. Failure to do so would create an inoperable system, as would be recognized as well known by a person of ordinary skill in the art. Applicants have not traversed this use of Official Notice and as such, it is regarded as admitted prior art. See MPEP 2144.03 (C).

Claim 8 recites what is generally known in the art as "garbage collection". Official notice is taken that Java™ and the Java™ virtual machine support garbage collection. It would have been obvious to a person of ordinary skill in the art at the time of Applicants' invention, in combination with his own knowledge of the particular art, to implement the system taught by Shimura using garbage collection because of the well-known advantages of garbage collection, such as ease of programming and recovery unused memory.

Claims 9-13 recite the server portion of the system of claims 1-5 and are rejected for the same reasons given above for claims 1-5.

Claims 14-17 recite the client portion of the system of claims 1 and 6-8 and are rejected for the same reasons given above for claims 1 and 6-8.

Claims 18-22 recite the methods performed by the system of claims 1-7 and are rejected for the same reasons given above for claims 1-7.

Claims 23-24 recite a computer program product and system according to claims 1-7 and are rejected for the same reasons given above for claim 1.

*Conclusion*


Any inquiry concerning this communication or earlier communications from the examiner should be directed to Jason Proctor whose telephone number is (571) 272-3713. The examiner can normally be reached on 8:30 am-4:30 pm M-F.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Leo Picard can be reached at (571) 272-3749. The fax phone number for the organization where this application or proceeding is assigned is (571) 273-8300.

Any inquiry of a general nature or relating to the status of this application should be directed to the TC 2100 Group receptionist: 571-272-2100. Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Jason Proctor  
Examiner  
Art Unit 2123

jsp

  
Paul L. Rodriguez 2/24/06  
Primary Examiner  
Art Unit 2125